CS415 Homework Assignment 3 : Threads

1 Problems

1.1 Problem 1

(Points: 20)List the reasons why a mode switch between threads may be less expensive computationally than a mode switch between processes.

1.2 Problem 2

(Points: 10) One of the design decisions required for the implementation of threads is what do you do when a program makes a system call. Once must decide if only the thread upon which the system call occurs blocks until system call returns or does the entire process block.

In an environment where there is a 1-to-1 mapping between user-level and kernel-level threads, explain why blocking only the thread that made the system call (and allowing the other threads in the process to continue) will make mutli-threaded program run faster than the matching single-threaded program on a machine that only has 1 core.

1.3 Problem 3

(Points: 40) Your Glorious Instructor (think the educational equivalent of a third-world dictator, with less hair) has an interesting problem for you to solve. You have two very large arrays in C++ with up to 10,000,000 data samples in each array. Your Glorious Instructor needs the average of the data stored in those arrays.

Write a function in C++ that returns void and accepts four parameters: a dataset, a dataset, a result set, and size:

```
void arrayAverage(int *dataset1, int* dataset2, int *resultset, int size)
```

After executing, the `resultset` will contain the result of pair-wise averaging matching elements of the two datasets (for instance `resultSet[0] = dataset1[0] + dataset2[0]`, and so on). Use threads to speed up this process... assume you have an 8-core machine. Partition the arrays based on the number of cores and assign different parts of two arrays to each thread. Use the thread join function to wait until all work is done.

1.4 Problem 4

(Points: 30) There are a class of algorithms known as Monte-Carlo algorithms that revolve around probability and random numbers to find a solution to a problem. A classic example of a Monte-Carlo Algorithm is used to estimate the value of π. In a plane, draw the unit square (a square whose length and width is 2 units centered on the origin $(0, 0)$). Then draw the unit circle inside this square where the unit circle is centered on $(0, 0)$ with a radius of 1 unit. We can estimate the value of π by randomly and uniformly generating a large number of points that lie inside one of the quadrants and then computing the ratio of the number of points that lie in the quarter circle to the total number of points generated. That number will be $\frac{\pi}{4}$.

Write a program that uses four threads to generate points in each of the four quadrants to estimate π. The first thread should work in the quadrant whose upper right corner is $(1, 1)$, the second in the quadrant $(-1, 1)$, the third in the quadrant $(-1, -1)$, and the fourth thread in $(1, -1)$. Each thread should be given a parameter for the number of points to use. Display the estimate of π generated by the each thread in your main thread after all threads have joined back to the main thread.

2 Submission instructions

You will need to document your submission in a short report that includes responses to questions and any supporting source code. Please attach this report, in PDF format, to your submission in Blackboard.